Government Smart Card Interoperability Specification v2.1 (NISTIR 6887, 2003 Edition)

Virtual Card Edge Interface File System Cards

Conformance Test Instantiation, Verification, and Reporting Scenarios

DRAFT

Alan Goldfine
February 10, 2005

This document contains the conformance test instantiation, verification, and reporting scenarios for the APDUs comprising the File System Virtual Card Edge Interface of version 2.1 of the Government Smart Card Interoperability Specification (GSC-IS), as contained in NIST Interagency Report 6887, 2003 Edition.

The 13 sections of this document correspond to the 13 sections in the File System Virtual Card Edge Interface Conformance Test Assertions document.

Appendix A contains the list of constant variables (symbolic constants) used in this document. A constant variable is indicated by a leading underscore (e.g., _PIN).

Appendix B contains a description of a card that would be sufficient for the testing of candidate File System Virtual Card Edge implementations, when using test suites built according to these scenarios.

APDUs are written with all 7 possible fields present, and with | used as the separator of the non-mandatory fields, i.e., in the form

CLA INS P1 P2|L_c|Data Field|L_e|

where some or all of L_c, Data Field, and L_e may be empty. For example,

00 B0 00 06| | |08 represents the APDU

00B0000608.

1. GET RESPONSE

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. Select the master file:

   Issue the following SELECT MASTER FILE APDU:

   ```
   00 A4 03 00|02|3F00| |
   ```

   Case 1: If the SELECT MASTER FILE returns SW1 SW2 == 90 00, then print

   "T=1 communications protocol. There is no test for GET RESPONSE."

   End current test.

   Case 2: If the SELECT MASTER FILE returns SW1 SW2 == 61 LL, then continue with step 2 of the Instantiation Scenario.

   Note: It is assumed that LL < FF.

   Case 3: If the SELECT MASTER FILE does not return either SW1 SW2 == 90 00 or SW1 SW2 == 61 LL, then print

   "The master file cannot be selected. GET RESPONSE cannot be tested."

Test for Assertion 1.1

The APDU is tested using valid parameters.

Instantiation Scenario

1. (Pre) Construct the starting state for GET RESPONSE.

2. Issue the following GET RESPONSE APDU:

   ```
   00 C0 00 00| | |LL|
   ```

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution)
   - the response message == the file control information.

Perform this verification by inspection.

Verification and Reporting Scenario:

1. **Case 1**: If the GET RESPONSE returns
   - SW1 SW2 == 90 00

   then manually inspect the response message.

   Case 1.1: If the response message == the file control information, then print

   "GET RESPONSE tested using valid parameters has been verified.

Status: Test 1.1 Passed."

Case 1.2: If the response message /= the file control information, then print
"GET RESPONSE tested using valid parameters has not been verified because the command did not return the correct data value.
Status: Test 1.1 Failed."

Case 2: If the GET RESPONSE APDU returns

- SW1 SW2 /= 90 00

then print
"GET RESPONSE tested using valid parameters has not been verified because the command did not return the return code.
Status: Test 1.1 Failed."

Test for Assertion 1.2

The APDU is tested using valid parameters, where the number of bytes specified to be read is less than the maximum available.

<u>Instantiation Scenario</u>

1. (Pre) Construct the starting state for GET RESPONSE.

2. Issue the following GET RESPONSE APDU:

```
00 C0 00 00|  |  |01|
```

<u>Verification Goal</u>:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 61 LL-1 (Normal processing with data still available)

<u>Verification and Reporting Scenario</u>:

1. **Case 1**: If the GET RESPONSE APDU returns
   - SW1 SW2 == 61 LL-1

   then print
   "GET RESPONSE tested using valid parameters, where the number of bytes specified to be read is less than the maximum available, has been verified.
   Status: Test 1.2 Passed."

   Case 2: If the GET RESPONSE APDU returns
   - SW1 SW2 /= 61 LL-1

   then print
   "GET RESPONSE tested using valid parameters, where the number of bytes specified to be read is less than the maximum available, has not been verified because the command did not return the correct status code.
   Status: Test 1.2 Failed."

Test for Assertion 1.3

The APDU is tested in the case where part of the data may be corrupted.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 1.4

The APDU is tested where the number of bytes specified to be read is greater than the maximum available.

Instantiation Scenario

1. (Pre) Construct the starting state for GET RESPONSE.

2. Issue the following GET RESPONSE APDU:

   00 C0 00 00| | |FF|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 67 00 (Wrong length) or 6C 04 (Wrong length)

Verification and Reporting Scenario:

1. **Case 1**: If the GET RESPONSE APDU returns
   - SW1 SW2 == 67 00 or 6C 04

   then print
   "GET RESPONSE where the number of bytes specified to be read is greater than the maximum available, has been verified.
   Status: Test 1.4 Passed."

   Case 2: If the GET RESPONSE APDU does not return
   - SW1 SW2 == 67 00 or 6C 04

   then print
   "GET RESPONSE where the number of bytes specified to be read is greater than the maximum available, has not been verified.
   Status: Test 1.4 Failed."

Test for Assertion 1.5

The APDU is tested using invalid parameters P1-P2.

Instantiation Scenario

1. (Pre) Construct the starting state for GET RESPONSE.

2. Issue the following GET RESPONSE APDU:

   00 C0 00 FE| | |04|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (Incorrect parameters P1P2)

Verification and Reporting Scenario:

1. **Case 1**: If the GET RESPONSE APDU returns
   - SW1 SW2 == 6A 86

then print

"GET RESPONSE tested using invalid parameters P1-P2 has been verified.
Status: Test 1.5 Passed."

Case 2: If the GET RESPONSE APDU does not return

- SW1 SW2 == 6A 86

then print

"GET RESPONSE tested using invalid parameters P1-P2 has not been verified.
Status: Test 1.5 Failed."

2. READ BINARY

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The master file contains an elementary file whose file ID is _goodEFileID1. All services of this file have access control rule ACR_ALWAYS. The file contains only the following four TLV records:
   - byte 00: 01
     byte 01: 04
     bytes 02-05: _existingValue1
   - byte 06: 02
     byte 07: 04
     bytes 08-0B: _existingValue2
   - byte 0C: 03
     byte 0D: 04
     bytes 0E-11: _existingValue3
   - byte 12: 04
     byte 13: 04
     bytes 14-17: _existingValue4.

   In particular, the file does not have a special length TLV (section 8.3 of the Spec.)

3. The master file contains a second elementary file whose file ID is _goodEFileID2. This file is PIN protected for all services, with the PIN == _goodPIN. The file contains the following TLV record:
   - byte 00: 11
     byte 01: 04
     bytes 02-05: _existingValue5.

   This file does not have a special length TLV.

4. Select the master file:

   Issue the following SELECT MASTER FILE APDU:
   00 A4 03 0C|02|3F00| |

   Case 1: If the SELECT MASTER FILE APDU returns SW1 SW2 == 90 00, then continue with Step 2 of the Instantiation Scenario.

   Case 2: If the SELECT MASTER FILE APDU does not return SW1 SW2 == 90 00, then print
   "The master file cannot be selected. READ BINARY cannot be tested."
   End current test.

Test for Assertion 2.1

The APDU is tested using valid parameters.

Instantiation Scenario

1. (Pre) Construct the starting state for READ BINARY.

2. (Pre) Select the elementary file whose file ID is _goodEFileID1:

   Issue the following SELECT EF UNDER SELECTED DF APDU:

   ```
   00 A4 02 0C|02|_goodEFileID1| |
   ```

   Case 1: If the SELECT EF UNDER SELECTED DF APDU returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT EF UNDER SELECTED DF APDU does not return SW1 SW2 == 90 00, then print
   "The elementary file cannot be selected. READ BINARY cannot be tested."
   End current test.

3. Issue the following READ BINARY APDU:

   ```
   00 B0 00 00| | |06|
   ```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution)
   - the response message ==
     byte 00: 01
     byte 01: 04
     bytes 02-05: _existingValue1.

Verification and Reporting Scenario:

1. **Case 1**: If the READ BINARY APDU returns the response code 90 00, then

   Case 1.1: If the response message ==
   byte 00: 01
   byte 01: 04
   bytes 02-05: _existingValue1

   then print
   "READ BINARY tested using valid parameters has been verified.
   Status: Test 2.1 Passed."

   Case 1.2: If the response message /=
   byte 00: 01
   byte 01: 04
   bytes 02-05: _existingValue1

   then print
   "READ BINARY tested using valid parameters has not been verified because an incorrect value was returned.
   Status: Test 2.1 Failed."

   Case 2: If the READ BINARY APDU does not return the response code 90 00 , then print

"READ BINARY tested using valid parameters did not return the correct status code.
Status: Test 2.1 Failed."

Test for Assertion 2.2

The APDU is tested in the case where part of the data may be corrupted.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 2.3

The APDU is tested such that the end of the specified file is reached before the specified number of bytes has been read.

Instantiation Scenario

1. (Pre) Construct the starting state for READ BINARY.

2. (Pre) Select the elementary file whose file ID is _goodEFileID1:

   Issue the following SELECT EF UNDER SELECTED DF APDU:

   00 A4 02 0C|02|_goodEFileID1| |

   Case 1: If the SELECT EF UNDER SELECTED DF APDU returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT EF UNDER SELECTED DF APDU does not return SW1 SW2 == 90 00, then print
   "The elementary file cannot be selected. READ BINARY cannot be tested."
   End current test.

3. Issue the following READ BINARY APDU:

   00 B0 00 00| | |20|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 62 82 (End of file reached), or 67 00 (Wrong length), or 6C XX (Wrong length, XX indicates the exact length) or 61 00.

Verification and Reporting Scenario:

1. **Case 1**: If the READ BINARY APDU returns any of the response codes 62 82, or 67 00, or 6C XX or 61 00, then print
   "READ BINARY, tested in the case where the end of the specified file is reached before the specified number of bytes has been read, has been verified.
   Status: Test 2.3 Passed."

   Case 2: If the READ BINARY APDU returns any response code other than 62 82, or 67 00, or 6C XX, or 61 00, then print

"READ BINARY, tested in the case where the end of the specified file is reached before the specified number of bytes has been read, has not been verified.
Status: Test 2.3 Failed."

Test for Assertion 2.4

The APDU is tested in the case where the command is incompatible with the file structure.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 2.5

The APDU is tested in the case where the security status of the selected file is not satisfied.

Instantiation Scenario

1. (Pre) Construct the starting state for READ BINARY.

2. (Pre) Select the elementary file whose file ID is _goodEFileID2:

   Issue the following SELECT EF UNDER SELECTED DF APDU:

   ```
   00 A4 02 0C|02|_goodEFileID2|  |
   ```

   Case 1: If the SELECT EF UNDER SELECTED DF APDU returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT EF UNDER SELECTED DF APDU does not return SW1 SW2 == 90 00, then print
   "The elementary file cannot be selected. READ BINARY cannot be tested."
   End current test.

Note: The security status of the selected file is deliberately not satisfied.

3. Issue the following READ BINARY APDU:

   ```
   00 B0 00 00|  |  |06|
   ```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 69 82 (Security status not satisfied).

Verification and Reporting Scenario:

1. **Case 1**: If the READ BINARY APDU returns the response code 69 82, then print
   "READ BINARY, tested in the case where necessary access control operations have not been completed, has been verified.
   Status: Test 2.5 Passed."

Case 2: If the READ BINARY APDU does not return the response code 69 82, then print
"READ BINARY, tested in the case where necessary access control operations have not been completed, has not been verified.
Status: Test 2.5 Failed."

Test for Assertion 2.6

The APDU is tested in the case where no elementary file on the card is currently selected.

Instantiation Scenario

1. (Pre) Construct the starting state for READ BINARY.

2. Issue the following READ BINARY APDU:

   00 B0 00 00| | |06|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 69 86 (Command not allowed) or 6A 82 (File not found).

Verification and Reporting Scenario:

1. **Case 1**: If the READ BINARY APDU returns either of the response codes 69 86 or 6A 82, then print
   "READ BINARY, tested in the case where no elementary file on the card is currently selected, has been verified.
   Status: Test 2.6 Passed."

   Case 2: If the READ BINARY APDU returns any response code other than 69 86 or 6A 82, then print
   "READ BINARY, tested in the case where no elementary file on the card is currently selected, has not been verified.
   Status: Test 2.6 Failed."

Test for Assertion 2.7

The APDU is tested in the case where function is not supported.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 2.8

The APDU is tested using an offset outside the selected file.

Instantiation Scenario

1. (Pre) Construct the starting state for READ BINARY.

2. (Pre) Select the elementary file whose file ID is _goodEFileID1:

   Issue the following SELECT EF UNDER SELECTED DF APDU:

```
00 A4 02 0C|02|_goodEFileID1| |
```

Case 1: If the SELECT EF UNDER SELECTED DF APDU returns SW1 SW2 == 90 00, then continue with Step 3.

Case 2: If the SELECT EF UNDER SELECTED DF APDU does not return SW1 SW2 == 90 00, then print
"The elementary file cannot be selected. READ BINARY cannot be tested."
End current test.

3. Issue the following READ BINARY APDU:

```
00 B0 00 20| | |06|
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6B 00 (Wrong parameters, offset outside the EF).

Verification and Reporting Scenario:

1. **Case 1**: If the READ BINARY APDU returns the response code 6B 00, then print
   "READ BINARY, tested in the case where the specified offset is outside the selected file, has been verified.
   Status: Test 2.8 Passed."

   Case 2: If the READ BINARY APDU returns any response code other than 6B 00, then print
   "READ BINARY, tested in the case where the specified offset is outside the selected file, has not been verified.
   Status: Test 2.8 Failed."

3. SELECT DF

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The card contains at least one dedicated file directly under the master file. The file ID for this file is _goodDFileID. This dedicated file contains at least one elementary file, whose file ID is _goodEFileID3. All services of this file have access control rule ACR_XAUTH.

3. The card does not contain a dedicated file with file ID _badDFileID directly under the master file.

4. Select the master file:

   Issue the following SELECT MASTER FILE APDU:

   ```
   00 A4 03 0C|02|3F00|  |
   ```

   Case 1: If the SELECT MASTER FILE APDU returns SW1 SW2 == 90 00, then continue with Step 2 of the Instantiation Scenario.

   Case 2: If the SELECT MASTER FILE APDU does not return SW1 SW2 == 90 00, then print
   "The master file cannot be selected. SELECT DF cannot be tested."
   End current test.

Note: The entry in Table 5-4 for SELECT DF specifies L_e to be empty; section 5.1.1.3 specifies L_e to be the number of bytes returned. The former makes sense for the case where no response is required; the latter makes sense for the case where a response is required. The Card Edge test suite uses this context sensitive distinction.

Test for Assertion 3.1

The APDU is tested using valid parameters (no response required).

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT DF.

2. Issue the following SELECT DF APDU:

   ```
   00 A4 01 0C|02|_goodDFileID|  |
   ```

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution)

   or
   - SW1 SW2 == 61 XX (Normal processing, data available to read).

Note: We allow for 61 XX, because some systems may generate a response even if none is requested.

2. The dedicated file identified by _goodDFileID is selected.

Perform this verification by attempting to select the elementary file whose ID is _goodEFileID3.

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT DF APDU returns the response code 90 00 or 61 XX, then verify that the dedicated file identified by _goodDFileID was indeed selected.

   Issue the following SELECT EF UNDER SELECTED DF APDU:

   00 A4 02 00|02 |_goodEFileID3| |

   Case 1.1: If the SELECT EF UNDER SELECTED DF APDU returns the response code 90 00 or 61 XX, then print
   "SELECT DF tested using valid parameters has been verified because a subsequent SELECT EF UNDER SELECTED DF was successful.
   Status: Test 3.1 Passed."

   Case 1.2: If the SELECT EF UNDER SELECTED DF APDU does not return the response code 90 00 or 61 XX, then print
   "SELECT DF tested using valid parameters has not been verified because a subsequent SELECT EF UNDER SELECTED DF was unsuccessful.
   Status: Test 3.1 Failed."

   Case 2: If the SELECT DF APDU does not return the response code 90 00 or 61 XX, then print
   "SELECT DF tested using valid parameters did not find the desired dedicated file.
   Status: Test 3.1 Failed."

Test for Assertion 3.2

The APDU is tested using valid parameters (response required).

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT DF.

2. Issue the following SELECT DF APDU:

   00 A4 01 00|02|_goodDFileID|03|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution)

   or
   - SW1 SW2 == 61 03 (Normal processing, data available to read).

Note: For systems using the T=0 communications protocol, the 61 XX response code is required by ISO 7816-4.

2. The dedicated file identified by _goodDFileID is selected.

Note: Future versions of the Card Edge test suite may also verify the file control information returned by this APDU.

Perform this verification by attempting to select the elementary file whose ID is _goodEFileID3.

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT DF APDU returns the response code 90 00 or 61 XX, then verify that the dedicated file identified by _goodDFileID was indeed selected.

   Issue the following SELECT EF UNDER SELECTED DF APDU:

   ```
   00 A4 02 00|02|_goodEFileID3|  |
   ```

   Case 1.1: If the SELECT EF UNDER SELECTED DF APDU returns the response code 90 00 or 61 XX, then print
   "SELECT DF tested using valid parameters has been verified because a subsequent SELECT EF UNDER SELECTED DF was successful.
   Status: Test 3.2 Passed."

   Case 1.2: If the SELECT EF UNDER SELECTED DF APDU does not return the response code 90 00 or 61 XX, then print
   "SELECT DF tested using valid parameters has not been verified because a subsequent SELECT EF UNDER SELECTED DF was unsuccessful.
   Status: Test 3.2 Failed."

   Case 2: If the SELECT DF APDU does not return the response code 90 00 or 61 XX, then print
   "SELECT DF tested using valid parameters did not find the desired dedicated file.
   Status: Test 3.2 Failed."

Test for Assertion 3.3

The APDU is tested using valid parameters (no response required) and a card in which the file to be selected has been deactivated.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 3.4

The APDU is tested using valid parameters (no response required) and a card in which the FCI is not formatted according to ISO 7816-4 Section 5.1.5.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 3.5

The APDU is tested where the function is not supported.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 3.6

The APDU is tested in the case where the file to be selected does not exist.

<u>Instantiation Scenario</u>

1. (Pre) Construct the starting state for SELECT DF.

2. Issue the following SELECT DF APDU:

```
00 A4 01 0C|02|_badDFileID|  |
```

<u>Verification Goal</u>:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 82 (File not found).

<u>Verification and Reporting Scenario</u>:

1. **Case 1**: If the SELECT DF APDU returns the response code 6A 82, then print
   "SELECT DF, tested in the case where the file to be selected does not exist, did not find the file to be selected.
   Status: Test 3.6 Passed."

   Case 2: If the SELECT DF APDU does not return the response code 6A 82, then print
   "SELECT DF, tested in the case where the file to be selected does not exist, returned an incorrect response code.
   Status: Test 3.6 Failed."

Test for Assertion 3.7

The APDU is tested using invalid parameters P1-P2.

<u>Instantiation Scenario</u>

1. (Pre) Construct the starting state for SELECT DF.

2. Issue the following SELECT DF APDU:

```
00 A4 01 FE|02|_goodDFileID|  |
```

<u>Verification Goal</u>:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (Incorrect parameters P1-P2).

<u>Verification and Reporting Scenario</u>:

1. **Case 1**: If the SELECT DF APDU returns the response code 6A 86, then print

"SELECT DF tested using invalid parameter P2 has been verified.
Status: Test 3.7 Passed."

Case 2: If the SELECT DF APDU does not return the response code 6A 86, then print

"SELECT DF tested using invalid parameter P2 returned an incorrect response code.
Status: Test 3.7 Failed."

Test for Assertion 3.8

The APDU is tested using an L_c inconsistent with P1-P2 (no response required).

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

4. SELECT EF UNDER SELECTED DF

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The card contains at least one dedicated file directly under the master file. The file ID for this file is _goodDFileID.

3. The dedicated file contains at least one elementary file, whose file ID is goodEFileID3. All services of this file have access control rule ACR_XAUTH. The file contains the following two TLV records:
   - byte 00: 21
     byte 01: 04
     bytes 02-05 == _existingValue6
   - byte 06: 22
     byte 07: 04
     bytes 08-0B == _existingValue7.

   The elementary file does not have a special length TLV.

4. The dedicated file does not contain an elementary file with file ID _badEFileID.

5. Select the master file:

   Issue the following SELECT MASTER FILE APDU:

   ```
   00 A4 03 0C|02|3F00|  |
   ```

   Case 1: If the SELECT MASTER FILE APDU returns SW1 SW2 == 90 00 or 61 XX, then continue with Step 6 of the Starting State.

   Case 2: If the SELECT MASTER FILE APDU does not return SW1 SW2 == 90 00 or 61 XX, then print
   "The master file cannot be selected. SELECT EF UNDER SELECTED DF cannot be tested."
   End current test.

6. Select the dedicated file, with File ID _goodDFileID, under the master file:

   Issue the following SELECT DF APDU:

   ```
   00 A4 01 0C|02|_goodDFileID|  |
   ```

   Case 1: If the SELECT DF APDU returns SW1 SW2 == 90 00 or 61 XX, then continue with Step 2 of the Instantiation Scenario.

   Case 2: If the SELECT DF APDU does not return SW1 SW2 == 90 00 or 61 XX, then print
   "The dedicated file under the master file cannot be selected. SELECT EF UNDER SELECTED DF cannot be tested."
   End current test.

Note: The entry in Table 5-4 for SELECT EF UNDER SELECTED DF specifies L_e to be empty; section 5.1.1.3.1 specifies L_e to be the number of bytes returned. The former makes sense for the case where no response is required; the latter makes sense for the case where a response is required. The Card Edge test suite uses this context sensitive distinction.

Test for Assertion 4.1

The APDU is tested using valid parameters (no response required).

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT EF UNDER SELECTED DF.

2. Issue the following SELECT EF UNDER SELECTED DF APDU:

```
00 A4 02 0C|02|_goodEFileID3|  |
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution)

   or
   - SW1 SW2 == 61 XX (Normal processing, data available to read).

Note: We allow for 61 XX, because some systems may generate a response even if none is requested.

2. The elementary file identified by _goodEFileID3 is selected.

Perform this verification by attempting to read the elementary file whose ID is _goodEFileID3.

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT EF UNDER SELECTED DF APDU returns the response code 90 00 or 61 XX, then verify that the elementary file identified by _goodEFileID3 was indeed selected.

   Issue the following READ BINARY APDU:

   ```
   00 B0 00 02|  |  |04|
   ```

   Case 1.1: If the READ BINARY APDU
   - returns the response code 90 00, and
   - response bytes 0-3 == _existingValue6

   then print
   "SELECT EF UNDER SELECTED DF tested using valid parameters has been verified because a subsequent READ BINARY was successful, indicating that the correct elementary file had been selected.
   Status: Test 4.1 Passed."

   Case 1.2: If the READ BINARY does not
   - does not return the response code 90 00, or
   - response bytes 0-3 /= _existingValue6

   then print

"SELECT EF UNDER SELECTED DF tested using valid parameters has not been verified because a subsequent READ BINARY was unsuccessful.
Status: Test 4.1 Failed."

Case 2: If the SELECT EF UNDER SELECTED DF APDU does not return the response code 90 00 or 61 XX, then print
"SELECT EF UNDER SELECTED DF tested using valid parameters did not select the desired elementary file.
Status: Test 4.1 Failed."

Test for Assertion 4.2

The APDU is tested using valid parameters (response required).

Instantiatio Scenario

1. (Pre) Construct the starting state for SELECT EF UNDER SELECTED DF.

2. Issue the following SELECT EF UNDER SELECTED DF APDU:

   00 A4 02 00|02|_goodEFileID3|03|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution)

   or
   - SW1 SW2 == 61 03 (Normal processing, data available to read).

Note: For systems using the T=0 communications protocol, the 61 XX response code is required by ISO 7816-4.

2. The elementary file identified by _goodEFileID3 is selected.

Note: Future versions of the Card Edge test suite may also verify the file control information returned by this APDU.

Perform this verification by attempting to read the elementary file whose ID is _goodEFileID3.

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT EF UNDER SELECTED DF APDU returns the response code 90 00 or 61 03, then verify that the elementary file identified by _goodEFileID3 was indeed selected.

   Issue the following READ BINARY APDU:

   00 B0 00 08| | |04|

   Case 1.1: If the READ BINARY APDU
   - returns the response code 90 00, and
   - response bytes 0-3 == _existingValue7

   then print
   "SELECT EF UNDER SELECTED DF tested using valid parameters has been verified because a subsequent READ BINARY was

successful, indicating that the correct elementary file had been selected.
Status: Test 4.2 Passed."

Case 1.2: If the READ BINARY does not

- does not return the response code 90 00, or
- response bytes 0-3 /= _existingValue7

then print

"SELECT EF UNDER SELECTED DF tested using valid parameters has not been verified because a subsequent READ BINARY was unsuccessful.
Status: Test 4.2 Failed."

Case 2: If the SELECT EF UNDER SELECTED DF APDU does not return the response code 90 00 or 61 03, then print

"SELECT EF UNDER SELECTED DF tested using valid parameters did not select the desired elementary file.
Status: Test 4.2 Failed."

Test for Assertion 4.3

The APDU is tested using valid parameters (no response required) and a card in which the file to be selected has been deactivated.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 4.4

The APDU is tested using valid parameters (no response required) and a card in which the FCI is not formatted according to ISO 7816-4 Section 5.1.5.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 4.5

The APDU is tested where the function is not supported.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 4.6

The APDU is tested in the case where the file to be selected does not exist.

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT EF UNDER SELECTED DF.

2. Issue the following SELECT EF UNDER SELECTED DF APDU:

   00 A4 02 0C|02|_badEFileID| |

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 82 (File not found).

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT EF UNDER SELECTED DF APDU returns the response code 6A 82, then print
   "SELECT EF UNDER SELECTED DF, tested in the case where the file to be selected does not exist, did not find the file to be selected.
   Status: Test 4.6 Passed."

   Case 2: If the SELECT EF UNDER SELECTED DF APDU does not return the response code 6A 82, then print
   "SELECT EF UNDER SELECTED DF, tested in the case where the file to be selected does not exist, returned an incorrect response code.
   Status: Test 4.6 Failed."

Test for Assertion 4.7

The APDU is tested using invalid parameters P1-P2.

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT EF UNDER SELECTED DF.

2. Issue the following SELECT EF UNDER SELECTED DF APDU:

   00 A4 00 FE|02|_goodDFileID| |

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (Incorrect parameters P1-P2).

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT EF UNDER SELECTED DF APDU returns the response code 6A 86, then print
   "SELECT EF UNDER SELECTED DF tested using invalid parameter P2 has been verified.
   Status: Test 4.7 Passed."

   Case 2: If the SELECT EF UNDER SELECTED DF APDU does not return the response code 6A 86, then print
   "SELECT EF UNDER SELECTED DF tested using invalid parameter P2 returned an incorrect response code.
   Status: Test 4.7 Failed."

Test for Assertion 4.8

The APDU is tested using an L_c inconsistent with P1-P2 (no response required).

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

5. SELECT FILE

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The card contains at least one dedicated file directly under the master file. The file ID for this file is _goodDFileID.

3. The dedicated file contains at least one elementary file, whose file ID is _goodEFileID3. All services of this file have access control rule ACR_XAUTH.

4. The elementary file contains the following two TLV records:
   - byte 00: 21
     byte 01: 04
     bytes 02-05 == _existingValue6
   - byte 06: 22
     byte 07: 04
     bytes 08-0B == _existingValue7.

   This file does not have a special length TLV.

5. The card does not contain a dedicated file with file ID _badDFileID directly under the master file.

6. Select the master file:

   Issue the following SELECT MASTER FILE APDU:
   00 A4 03 0C|02|3F00| |

   Case 1: If the SELECT MASTER FILE APDU returns SW1 SW2 == 90 00 or 61 XX, then continue with 7.

   Case 2: If the SELECT MASTER FILE APDU does not return SW1 SW2 == 90 00 or 61 XX, then print
   "The master file cannot be selected. SELECT FILE cannot be tested."
   End current test.

7. Select the CCC:

   Issue the following SELECT EF UNDER SELECTED DF APDU:
   00 A4 02 0C|02|DB00| |

   Case 1: If the SELECT EF UNDER SELECTED DF returns SW1 SW2 == 90 00 or 61 XX, then continue with Step 2 of the Instantiation Scenario.

   Case 2: If the SELECT EF UNDER SELECTED DF does not return SW1 SW2 == 90 00 or 61 XX, then print
   "The CCC cannot be selected. SELECT FILE cannot be tested."
   End current test.

Note: Section 5.1.1.4 specifies L_e to be empty. However, in the case where a response is required, L_e must specify the number of bytes returned in the response.

Test for Assertion 5.1

The APDU is tested using valid parameters (explicit selection, no response required).

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT FILE.

2. Issue the following SELECT FILE APDU:

```
        00 A4 00 0C|02|_goodEFileID3| |
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution)

   or
   - SW1 SW2 == 61 XX (Normal processing, data available to read).

Note: We allow for 61 XX, because some systems may generate a response even if none is requested.

2. The elementary file identified by _goodEFileID3 is selected.

Perform this verification by attempting to read the elementary file whose ID is _goodEFileID3.

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT FILE APDU returns the response code 90 00 or 61 XX, then verify that the elementary file identified by _goodEFileID3 was indeed selected.

   Issue the following READ BINARY APDU:

   ```
           00 B0 00 00| | |04|
   ```

   Case 1.1: If the READ BINARY APDU returns the response code 90 00, then

   Case 1.1.1: If the response message ==
   byte 00 == 21
   byte 01 == 04
   - bytes 02-05 == _existingValue6

   then print
   "SELECT FILE tested using valid parameters (explicit selection, no response required) has been verified because a subsequent READ BINARY was successful, indicating that the correct file had been selected.
   Status: Test 5.1 Passed."

   Case 1.1.2: If it is not true that the response message ==
   byte 00 == 21
   byte 01 == 04

```
    bytes 02-05 == _existingValue6
  then print
    "SELECT tested using valid parameters (explicit selection,
    no response required) has not been verified because a
    subsequent READ BINARY returned an incorrect value.
    Status: Test 5.1 Failed."
```

Case 1.2: If the READ BINARY does not return the response code 90 00, then print

"SELECT tested using valid parameters (explicit selection, no response required) has not been verified because a subsequent READ BINARY returned an incorrect response code.
Status: Test 5.1 Failed."

Case 2: If the SELECT FILE APDU does not return the response code 90 00 or 61 XX, then print

"SELECT FILE tested using valid parameters (explicit selection, no response required) did not select the desired file.
Status: Test 5.1 Failed."

Test for Assertion 5.2

The APDU is tested using valid parameters (explicit selection, response required).

Instantiatio Scenario

1. (Pre) Construct the starting state for SELECT FILE.

2. Issue the following SELECT FILE APDU:

   00 A4 00 00|02|_goodEFileID3|03|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution)

   or
   - SW1 SW2 == 61 03 (Normal processing, data available to read).

Note: For systems using the T=0 communications protocol, the 61 XX response code is required by ISO 7816-4.

2. The elementary file identified by _goodEFileID3 is selected.

Note: Future versions of the Card Edge test suite may also verify the file control information returned by this APDU.

Perform this verification by attempting to read the elementary file whose ID is _goodEFileID3.

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT FILE APDU returns the response code 90 00 or 61 XX, then verify that the elementary file identified by _goodEFileID3 was indeed selected.

   Issue the following READ BINARY APDU:

00 B0 00 06| | |06|

Case 1.1: If the READ BINARY APDU returns the response code 90 00, then

Case 1.1.1: If the response message ==
byte 06 == 22
byte 07 == 04
bytes 08-0B == _existingValue7
then print
"SELECT FILE tested using valid parameters (explicit selection, no response required) has been verified because a subsequent READ BINARY was successful, indicating that the correct file had been selected.
Status: Test 5.2 Passed."

Case 1.1.2: If it is not true that the response message ==
byte 06 == 22
byte 07 == 04
bytes 08-0B == _existingValue7
then print
"SELECT tested using valid parameters (explicit selection, no response required) has not been verified because a subsequent READ BINARY returned an incorrect value.
Status: Test 5.2 Failed."

Case 1.2: If the READ BINARY does not return the response code 90 00, then print
"SELECT tested using valid parameters (explicit selection, no response required) has not been verified because a subsequent READ BINARY returned an incorrect response code.
Status: Test 5.2 Failed."

Case 2: If the SELECT FILE APDU does not return the response code 90 00 or 61 03, then print
"SELECT FILE tested using valid parameters (explicit selection, response required) did not select the desired file.
Status: Test 5.2 Failed."

Test for Assertion 5.3

The APDU is tested using valid parameters (select parent DF, no response required).

Instantiation Scenario
1. (Pre) Construct the starting state for SELECT FILE.

2. Issue the following SELECT FILE APDU:
00 A4 03 0C| | | |

Verification Goal:
To verify the expected results:

1. The APDU returns
    - SW1 SW2 == 90 00 (Successful execution)

   or
    - SW1 SW2 == 61 03 (Normal processing, data available to read).

Note: For systems using the T=0 communications protocol, the 61 XX response code is required by ISO 7816-4.

2. The Master File is selected.

Perform this verification by attempting to select the dedicated file whose ID is _goodDFileID.

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT FILE APDU returns the response code 90 00 or 61 XX, then verify that the master file was indeed selected.

   Issue the following SELECT DF APDU:

   ```
   00 A4 02 00|02|_goodDFileID|  |
   ```

   Case 1.1: If the SELECT DF APDU returns the response code 90 00 or 61 XX, then print
   "SELECT FILE tested using valid parameters (select parent DF, no response required) has been verified because a subsequent SELECT DF was successful.
   Status: Test 5.3 Passed."

   Case 1.2: If the SELECT DF APDU does not return the response code 90 00 or 61 XX, then print
   "SELECT FILE tested using valid parameters (select parent DF, no response required) has not been verified because a subsequent SELECT DF was unsuccessful.
   Status: Test 5.3 Failed."

   Case 2: If the SELECT FILE APDU does not return the response code 90 00 or 61 XX, then print
   "SELECT FILE tested using valid parameters (select parent DF, no response required) returned an incorrect response code.
   Status: Test 5.3 Failed."

Test for Assertion 5.4

The APDU is tested using valid parameters (select parent DF, response required).

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT FILE.

2. Issue the following SELECT FILE APDU:

   ```
   00 A4 03 00|00|  |03|
   ```

Verification Goal:

To verify the expected results:

1. The APDU returns
    - SW1 SW2 == 90 00 (Successful execution)

or

- SW1 SW2 == 61 03 (Normal processing, data available to read).

Note: For systems using the T=0 communications protocol, the 61 XX response code is required by ISO 7816-4.

2. The Master File is selected.

Note: Future versions of the Card Edge test suite may also verify the file control information returned by this APDU.

Perform this verification by attempting to select the dedicated file whose ID is _goodDFileID.

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT FILE APDU returns the response code 90 00 or 61 XX, then verify that the master file was indeed selected.

   Issue the following SELECT DF APDU:

   00 A4 02 00|02|_goodDFileID| |

   Case 1.1: If the SELECT DF APDU returns the response code 90 00 or 61 XX, then print
   "SELECT FILE tested using valid parameters (select parent DF, response required) has been verified because a subsequent SELECT DF was successful.
   Status: Test 5.4 Passed."

   Case 1.2: If the SELECT DF APDU does not return the response code 90 00 or 61 XX, then print
   "SELECT FILE tested using valid parameters (select parent DF, response required) has not been verified because a subsequent SELECT DF was unsuccessful.
   Status: Test 5.4 Failed."

   Case 2: If the SELECT FILE APDU does not return the response code 90 00 or 61 XX, then print
   "SELECT FILE tested using valid parameters (select parent DF, response required) returned an incorrect response code.
   Status: Test 5.4 Failed."

Test for Assertion 5.5

The APDU is tested using valid parameters (explicit selection, no response required) and a card in which the file to be selected has been deactivated.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 5.6

The APDU is tested using valid parameters (explicit selection, no response required) and a card in which the FCI is not formatted according to ISO 7816-4 Section 5.1.5.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 5.7

The APDU is tested where the function is not supported.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 5.8

The APDU is tested using valid parameters (explicit selection, no response required) and a card that does not contain a file identified by the specified file ID.

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT FILE.

2. Issue the following SELECT FILE APDU:

   00 A4 00 0C|02|_badDFileID| |

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 82 (File not found).

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT FILE APDU returns the response code 6A 82, then print
   "SELECT FILE, tested in the case where the file to be selected does not exist, did not find the file to be selected.
   Status: Test 5.8 Passed."

   Case 2: If the SELECT FILE APDU does not return the response code 6A 82, then print
   "SELECT FILE, tested in the case where the file to be selected does not exist, returned an incorrect response code.
   Status: Test 5.8 Failed."

Test for Assertion 5.9

The APDU is tested using invalid parameters P1-P2.

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT FILE.

2. Issue the following SELECT FILE APDU:

    00 A4 00 FE|02|_goodDFileID| |

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (Incorrect parameters P1-P2).

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT FILE APDU returns the response code 6A 86, then print
   "SELECT FILE tested using invalid parameter P2 has been verified.
   Status: Test 5.9 Passed."

   Case 2: If the SELECT FILE APDU does not return the response code 6A 86, then print
   "SELECT FILE tested using invalid parameter P2 returned an incorrect response code.
   Status: Test 5.9 Failed."

Test for Assertion 5.10

The APDU is tested using an L_c inconsistent with P1-P2 (explicit selection, no response required).

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

6. SELECT MASTER FILE

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. There is currently no file selected.

Note: Version 2 of the Card Edge test suite may have one or more additional tests for each assertion. These tests would start with a file already selected, and then issue the APDU

Note: Section 5.1.1.5 specifies L_e to be empty. However, in the case where a response is required, L_e must specify the number of bytes returned in the response.

Test for Assertion 6.1

The APDU is tested using valid parameters (no response required).

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT MASTER FILE.

2. Issue the following SELECT MASTER FILE APDU:

```
00 A4 03 0C|02|3F00|  |
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution)

   or
   - SW1 SW2 == 61 XX (Normal processing, data available to read).

Note: We allow for 61 XX, because some systems may generate a response even if none is requested.

2. The master file is selected.

Perform this verification by attempting to select the CCC.

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT MASTER FILE APDU returns the response code 90 00 or 61 XX, then verify that the master file was indeed selected.

   Issue the following SELECT EF UNDER SELECTED DF APDU:

   ```
   00 A4 02 00|02|DB00|  |
   ```

   Case 1.1: If the SELECT EF UNDER SELECTED DF APDU returns the response code 90 00 or 61 XX, then print
   "SELECT MASTER FILE tested using valid parameters has been verified because a subsequent SELECT EF UNDER SELECTED DF to retrieve the CCC was successful.
   Status: Test 6.1 Passed."

Case 1.2: If the SELECT EF UNDER SELECTED DF APDU does not return the response code 90 00 or 61 XX, then print

"SELECT MASTER FILE tested using valid parameters has not been verified because a subsequent SELECT EF UNDER SELECTED DF to retrieve the CCC was unsuccessful.
Status: Test 6.1 Failed."

Case 2: If the SELECT MASTER FILE APDU does not return the response code 90 00 or 61 XX, then print

"SELECT MASTER FILE tested using valid parameters returned an incorrect response code.
Status: Test 6.1 Failed."

Test for Assertion 6.2

The APDU is tested using valid parameters (response required).

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT MASTER FILE.

2. Issue the following SELECT MASTER FILE APDU:

```
        00 A4 03 00|02|3F00|03|
```

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution)

   or
   - SW1 SW2 == 61 03 (Normal processing, data available to read).

Note: For systems using the T=0 communications protocol, the 61 XX response code is required by ISO 7816-4.

2. The master file is selected.

Note: Future versions of the Card Edge test suite may also verify the file control information returned by this APDU.

Perform this verification by attempting to select the CCC.

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT MASTER FILE APDU returns the response code 90 00 or 61 03, then verify that the master file was indeed selected.

   Issue the following SELECT EF UNDER SELECTED DF APDU:

```
        00 A4 02 00|02|DB00|  |
```

   Case 1.1: If the SELECT EF UNDER SELECTED DF APDU returns the response code 90 00 or 61 03, then print

   "SELECT MASTER FILE tested using valid parameters has been verified because a subsequent SELECT EF UNDER SELECTED DF to retrieve the CCC was successful.
   Status: Test 6.2 Passed."

Case 1.2: If the SELECT EF UNDER SELECTED DF APDU does not return the response code 90 00 or 61 03, then print
"SELECT MASTER FILE tested using valid parameters has not been verified because a subsequent SELECT EF UNDER SELECTED DF to retrieve the CCC was unsuccessful.
Status: Test 6.2 Failed."

Case 2: If the SELECT MASTER FILE APDU does not return the response code 90 00 or the response code 61 03, then print
"SELECT MASTER FILE tested using valid parameters returned an incorrect response code.
Status: Test 6.2 Failed."

Test for Assertion 6.3

The APDU is tested using valid parameters (no response required) and a card in which the file to be selected has been deactivated.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 6.4

The APDU is tested using valid parameters (no response required) and a card in which the FCI is not formatted according to ISO 7816-4 Section 5.1.5.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 6.5

The APDU is tested where the function is not supported.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 6.6

The APDU is tested using invalid parameters P1-P2.

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT MASTER FILE.

2. Issue the following SELECT MASTER FILE APDU:
   00 A4 03 FE|02|3F00| |

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (Incorrect parameters P1-P2).

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT MASTER FILE APDU returns the response code 6A 86, then print

   "SELECT MASTER FILE tested using invalid parameter P2 has been verified.
   Status: Test 6.6 Passed."

   Case 2: If the SELECT MASTER FILE APDU does not return the response code 6A 86, then print

   "SELECT MASTER FILE tested using invalid parameter P2 returned an incorrect response code.
   Status: Test 6.6 Failed."

Test for Assertion 6.7

The APDU is tested using an L_c inconsistent with P1-P2 (no response required).

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

7. UPDATE BINARY

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The master file contains an elementary file whose file ID is _goodEFileID1. All services of this file have access control rule ACR_ALWAYS. The file does not have a special length TLV (section 8.3 of the Spec.) The file contains only the following four TLV records:
    - byte 00: 01
      byte 01: 04
      bytes 02-05: _existingValue1
    - byte 06: 02
      byte 07: 04
      bytes 08-0B: _existingValue2
    - byte 0C: 03
      byte 0D: 04
      bytes 0E-11: _existingValue3
    - byte 12: 04
      byte 13: 04
      bytes 14-17: _existingValue4.

    The file has a maximum size of 17 bytes.

3. The master file contains a second elementary file whose file ID is _goodEFileID2. This file is PIN protected for all services, with the PIN == _goodPIN. The file contains the following TLV record:
    - byte 00: 11
      byte 01: 04
      bytes 02-05: _existingValue5.

    This file does not have a special length TLV.

4. Select the master file:

    Issue the following SELECT MASTER FILE APDU:

    00 A4 03 0C|02|3F00| |

    Case 1: If the SELECT MASTER FILE APDU returns SW1 SW2 == 90 00 or 61 XX, then continue with Step 2 of the Instantiation Scenario.

    Case 2: If the SELECT MASTER FILE APDU does not return SW1 SW2 == 90 00 or 61 XX, then print
    "The master file cannot be selected. UPDATE BINARY cannot be tested."
    End current test.

Test for Assertion 7.1

The APDU is tested using valid parameters.

Instantiation Scenario

1. (Pre) Construct the starting state for UPDATE BINARY.

2. (Pre) Select the elementary file whose file ID is _goodEFileID1:

   Issue the following SELECT EF UNDER SELECTED DF APDU:

   00 A4 02 0C|02|_goodEFileID1| |

   Case 1: If the SELECT EF UNDER SELECTED DF APDU returns SW1 SW2 == 90 00 or 61 XX, then continue with Step 3.

   Case 2: If the SELECT EF UNDER SELECTED DF APDU does not return SW1 SW2 == 90 00 or 61 XX, then print
   "The elementary file cannot be selected. UPDATE BINARY cannot be tested."
   End current test.

3. Issue the following UPDATE BINARY APDU:

   00 D6 00 02|04|_newValue1| |

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution) or 63 CX (Successful updating after X retries)

2. The block of 4 bytes specified to be updated now has the value _newValue1.

Verification and Reporting Scenario:

1. **Case 1**: If the UPDATE BINARY APDU returns the response code 90 00 or the code 63 CX, then

   Issue the following READ BINARY APDU:

   00 B0 00 02| | |04|

   Case 1.1: If the 4 byte data field in the response message of the READ BINARY == _newValue1, then print
   "UPDATE BINARY tested using valid parameters has been verified.
   Status: Test 7.1 Passed."

   Case 1.2: If the 4 byte data field in the response message of the READ BINARY /= _newValue1, then print
   "UPDATE BINARY tested using valid parameters has not been verified because the data was not updated correctly.
   Status: Test 7.1 Failed."

   Case 2: If the UPDATE BINARY APDU does not return either of the response codes 90 00 or 63 CX, then print
   "UPDATE BINARY tested using valid parameters did not return the correct status code.
   Status: Test 7.1 Failed."

Test for Assertion 7.2

The APDU is tested such that the specified number of bytes to be updated is too large to be accommodated by the selected file.

Instantiation Scenario

1. (Pre) Construct the starting state for UPDATE BINARY.

2. (Pre) Select the elementary file whose file ID is _goodEFileID1:

   Issue the following SELECT EF UNDER SELECTED DF APDU:

   ```
   00 A4 02 0C|02|_goodEFileID1| |
   ```

   Case 1: If the SELECT EF UNDER SELECTED DF APDU returns SW1 SW2 == 90 00 or 61 XX, then continue with Step 3.

   Case 2: If the SELECT EF UNDER SELECTED DF APDU does not return SW1 SW2 == 90 00 or 61 XX, then print
   "The elementary file cannot be selected. UPDATE BINARY cannot be tested."
   End current test.

3. Issue the following UPDATE BINARY APDU:

   ```
   00 D6 00 16|04|_newValue2| |
   ```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 67 00 (Wrong length)

2. No changes are made to the selected file.

Verification and Reporting Scenario:

1. **Case 1**: If the UPDATE BINARY APDU returns the response code 67 00, then

   Issue the following READ BINARY APDU:

   ```
   00 B0 00 14| | |04|
   ```

   Case 1.1: If the 4 byte data field in the response message of the READ BINARY == _existingValue4, then print
   "UPDATE BINARY, tested where the specified number of bytes to be updated is too large to be accommodated by the selected file, has been verified.
   Status: Test 7.2 Passed."

   Case 1.2: If the 4 byte data field in the response message of the READ BINARY /= _existingValue4, then print
   "UPDATE BINARY, tested where the specified number of bytes to be updated is too large to be accommodated by the selected file, has not been verified because the original data had been changed.
   Status: Test 7.2 Failed."

Case 2: If the UPDATE BINARY APDU does not return the response codes 67 00, then print

"UPDATE BINARY, tested where the specified number of bytes to be updated is too large to be accommodated by the selected file, did not return the correct status code.
Status: Test 7.2 Failed."

Test for Assertion 7.3

The APDU is tested in the case where the command is incompatible with the file structure.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 7.4

The APDU is tested in the case where the security status of the selected file is not satisfied.

Instantiation Scenario

1. (Pre) Construct the starting state for UPDATE BINARY.

2. ((Pre) Select the elementary file whose file ID is _goodEFileID2:

   Issue the following SELECT EF UNDER SELECTED DF APDU:

   00 A4 02 0C|02|_goodEFileID2| |

   Case 1: If the SELECT EF UNDER SELECTED DF APDU returns SW1 SW2 == 90 00 or 61 XX, then continue with Step 3.

   Case 2: If the SELECT EF UNDER SELECTED DF APDU does not return SW1 SW2 == 90 00 or 61 XX, then print

   "The elementary file cannot be selected. UPDATE BINARY cannot be tested."
   End current test.

Note: The security status of the selected file is deliberately not satisfied.

3. Issue the following UPDATE BINARY APDU:

   00 D6 00 0C|04|_newValue3| |

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 69 82 (Security status not satisfied)

2. No changes are made to the selected file.

Verification and Reporting Scenario:

1. **Case 1**: If the UPDATE BINARY APDU returns the response code 69 82, then

Issue the following READ BINARY APDU:

```
        00 B0 00 0C|  |  |04|
```

Case 1.1: If the 4 byte data field in the response message of the READ BINARY == _existingValue3, then print

"UPDATE BINARY, tested in the case where the security status of the selected file is not satisfied, has been verified.
Status: Test 7.4 Passed."

Case 1.2: If the 4 byte data field in the response message of the READ BINARY /= _existingValue3, then print

"UPDATE BINARY, tested in the case where the security status of the selected file is not satisfied, has not been verified because the original data had been changed.
Status: Test 7.4 Failed."

Case 2: If the UPDATE BINARY APDU does not return the response codes 69 82, then print

"UPDATE BINARY, tested in the case where the security status of the selected file is not satisfied, did not return the correct status code.
Status: Test 7.4 Failed."

Test for Assertion 7.5

The APDU is tested in the case where no elementary file on the card is currently selected.

Instantiation Scenario

1. (Pre) Construct the starting state for UPDATE BINARY.

2. Issue the following UPDATE BINARY APDU:

```
        00 D6 00 00|00|_newValue4|  |
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 69 86 (Command not allowed) or 6A 82 (File not found).

Verification and Reporting Scenario:

1. **Case 1**: If the UPDATE BINARY APDU returns either of the response codes 69 86 or 6A 82, then print

   "UPDATE BINARY, tested in the case where no elementary file on the card is currently selected, has been verified.
   Status: Test 7.5 Passed."

   Case 2: If the UPDATE BINARY APDU returns any response code other than 69 86 or 6A 82, then print

   "READ BINARY, tested in the case where no elementary file on the card is currently selected, has not been verified.
   Status: Test 7.5 Failed."

Test for Assertion 7.6

The APDU is tested in the case where function is not supported.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 7.7

The APDU is tested using an offset outside the selected file.

Instantiation Scenario

1. (Pre) Construct the starting state for UPDATE BINARY.

2. (Pre) Select the elementary file whose file ID is _goodEFileID1:

   Issue the following SELECT EF UNDER SELECTED DF APDU:

   ```
   00 A4 02 0C|02| _goodEFileID1|  |
   ```

   Case 1: If the SELECT EF UNDER SELECTED DF APDU returns SW1 SW2 == 90 00 or 61 XX, then continue with Step 3.

   Case 2: If the SELECT EF UNDER SELECTED DF APDU does not return SW1 SW2 == 90 00 or 61 XX, then print
   "The elementary file cannot be selected. READ BINARY cannot be tested."
   End current test.

3. Issue the following UPDATE BINARY APDU:

   ```
   00 D6 00 30|04|_newValue5|  |
   ```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6B 00 (Wrong parameters).

Verification and Reporting Scenario:

1. **Case 1**: If the UPDATE BINARY APDU returns the response code 6B 00, then print
   "UPDATE BINARY, tested in the case where the specified offset is outside the selected file, has been verified.
   Status: Test 7.7 Passed."

   Case 2: If the UPDATE BINARY APDU returns any response code other than 6B 00, then print
   "UPDATE BINARY, tested in the case where the specified offset is outside the selected file, has not been verified.
   Status: Test 7.7 Failed."

8. EXTERNAL AUTHENTICATE

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The master file contains an elementary file whose file ID is _goodEFileID4. All services of this file have access control rule ACR_XAUTH. The file contains the following TLV record:
   - byte 00: 41
     byte 01: 04
     bytes 02-05: _existingValue8.

   This file does not have a special length TLV.

3. Select the master file:

   Issue the following SELECT MASTER FILE APDU:
   00 A4 03 0C|02|3F00| |

   Case 1: If the SELECT MASTER FILE APDU returns SW1 SW2 == 90 00 or 61 XX, then continue with Step 4.

   Case 2: If the SELECT MASTER FILE APDU does not return SW1 SW2 == 90 00 or 61 XX, then print
   "The master file cannot be selected. EXTERNAL AUTHENTICATE cannot be tested."
   End current test.

4. Select the elementary file whose file ID is _goodEfileID4:

   Issue the following SELECT EF UNDER SELECTED DF APDU:
   00 A4 02 0C|02|_goodEFileID4| |

   Case 1: If the SELECT EF UNDER SELECTED DF APDU returns SW1 SW2 == 90 00 or 61 XX, then continue with Step 2 of the Instantiation Scenario.

   Case 2: If the SELECT EF UNDER SELECTED DF APDU does not return SW1 SW2 == 90 00 or 61 XX, then print
   "The elementary file cannot be selected. EXTERNAL AUTHENTICATE cannot be tested."
   End current test.

5. Verify that the selected file does not allow read access:

   Issue the following READ BINARY APDU:
   00 B0 00 02| | |04|

   Case 1: If the READ BINARY returns SW1 SW2 == 69 82, then continue with Step 6.

   Case 2: If the READ BINARY does not return SW1 SW2 == 69 82, then print

"It cannot be verified that the currently selected file does not allow read access. EXTERNAL AUTHENTICATE cannot be tested."

End current test.

6. Issue the following GET CHALLENGE APDU:

   00 84 00 00| | |08|

   Case 1: If the GET CHALLENGE APDU returns SW1 SW2 == 90 00,

   Let challenge0800 == the 8 byte data field in the response message.

   Encrypt challenge0800, producing cryptogram0800.

   Continue with Step 2 of the Instantiation Scenario.

   Case 2: If the GET CHALLENGE APDU does not return SW1 SW2 == 90 00, then print

   "GET CHALLENGE was not successful. EXTERNAL AUTHENTICATE cannot be tested."

   End current test.

Test for Assertion 8.1

The APDU is tested using valid parameters.

Instantiation Scenario

1. (Pre) Construct the starting state for EXTERNAL AUTHENTICATE.

2. Issue the following EXTERNAL AUTHENTICATE APDU:

   00 82 00 00|08|cryptogram0800| |

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution).

2. The currently selected file is now open to read access.

Perform this verification by attempting to read the currently selected file.

Verification and Reporting Scenario:

1. **Case 1**: If the EXTERNAL AUTHENTICATE APDU returns SW1 SW2 == 90 00, then issue the following READ BINARY APDU:

   00 B0 00 02| | |04|

   Case 1.1: If the READ BINARY returns SW1 SW2 == 90 00, then print

   "EXTERNAL AUTHENTICATE tested using valid parameters has been verified.
   Status: Test 8.1 Passed."

Case 1.2: If the READ BINARY does not return SW1 SW2 == 90 00, then print

"EXTERNAL AUTHENTICATE tested using valid parameters has not been verified because a subsequent READ BINARY was rejected.
Status: Test 8.1 Failed."

Case 2: If the EXTERNAL AUTHENTICATE APDU does not return SW1 SW2 == 90 00, then print

"EXTERNAL AUTHENTICATE tested using valid parameters did not return the correct status code.
Status: Test 8.1 Failed."

Test for Assertion 8.2

The APDU is tested using an invalid cryptogram.

Instantiation Scenario

1. (Pre) Construct the starting state for EXTERNAL AUTHENTICATE.

2. Issue the following EXTERNAL AUTHENTICATE APDU:

```
00 82 00 00|08|_badCryptogram| |
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 63 00 (No information given), 63 CX (Authentication failed), or 69 83 (Authentication method blocked).

2. The currently selected file is still not open to read access.

Perform this verification by attempting to read the currently selected file.

Verification and Reporting Scenario:

1. **Case 1**: If the EXTERNAL AUTHENTICATE APDU returns SW1 SW2 == 63 00, 63 CX, or 69 83, then issue the following READ BINARY APDU:

```
00 B0 00 02| | |04|
```

Case 1.1: If the READ BINARY does return SW1 SW2 == 69 82 (security status not satisfied), then print

"EXTERNAL AUTHENTICATE tested using a bad cryptogram has been verified.
Status: Test 8.2 Passed."

Case 1.2: If the READ BINARY returns SW1 SW2 == 90 00, then print

"EXTERNAL AUTHENTICATE tested using a bad cryptogram has not been verified because a subsequent READ BINARY was successful.
Status: Test 8.2 Failed."

Case 2: If the EXTERNAL AUTHENTICATE APDU does not return SW1 SW2 == 63 00, 63 CX, or 69 83, then print

"EXTERNAL AUTHENTICATE tested using a bad cryptogram did not return the correct status code.

Status: Test 8.2 Failed."

Test for Assertion 8.3

The APDU is tested using a bad data field length.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 8.4

The APDU is tested in the case where the referenced data is deactivated.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 8.5

The APDU is tested in a context where the command is not allowed.

Note: This is interpreted to mean that EXTERNAL AUTHENTICATE must immediately follow a GET CHALLENGE.

Instantiatio Scenario

1. (Pre) Construct the starting state for EXTERNAL AUTHENTICATE.

Need to insert a "dummy" command here to ensure that the EXTERNAL AUTHENTICATE does not immediately follow a GET CHALLENGE.

2. (Pre) Issue the following READ BINARY APDU:

   00 B0 00 02| | |04|

2. Issue the following EXTERNAL AUTHENTICATE APDU:

   00 82 00 00|08|cryptogram0800| |

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 69 85 (Conditions of use not satisfied), 63 00 (No information given), 63 CX (Authentication failed), or 69 83 (Authentication method blocked).

2. The currently selected file is still not open to read access.

Perform this verification by attempting to read the currently selected file.

Verification and Reporting Scenario:

1. **Case 1**: If the EXTERNAL AUTHENTICATE APDU returns SW1 SW2 == 63 00, 63 CX, or 69 83, then issue the following READ BINARY APDU:

   00 B0 00 02| | |04|

Case 1.1: If the READ BINARY does return SW1 SW2 == 6982 (security status not satisfied), then print

"EXTERNAL AUTHENTICATE tested in a context where the command is not allowed has been verified.
Status: Test 8.5 Passed."

Case 1.2: If the READ BINARY returns SW1 SW2 == 90 00, then print

"EXTERNAL AUTHENTICATE tested in a context where the command is not allowed has not been verified because a subsequent READ BINARY was successful.
Status: Test 8.5 Failed."

Case 2: If the EXTERNAL AUTHENTICATE APDU does not return SW1 SW2 == 63 00, 63 00, 63 CX, or 69 83, then print

"EXTERNAL AUTHENTICATE tested in a context where the command is not allowed did not return the correct status code.
Status: Test 8.5 Failed."

Test for Assertion 8.6

The APDU is tested using invalid parameters P1-P2.

Instantiation Scenario

1. (Pre) Construct the starting state for EXTERNAL AUTHENTICATE.

2. Issue the following EXTERNAL AUTHENTICATE APDU:

```
00 82 00 FE|08|cryptogram0800| |
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (Incorrect parameters P1-P2) or 6A 88 (Referenced data not found).

2. The currently selected file is still not open to read access.

Perform this verification by attempting to read the currently selected file.

Verification and Reporting Scenario:

1. **Case 1**: If the EXTERNAL AUTHENTICATE APDU returns SW1 SW2 == 6A 86 or 6A 88, then issue the following READ BINARY APDU:

```
00 B0 00 02| | |04|
```

Case 1.1: If the READ BINARY does not return SW1 SW2 == 90 00, then print

"EXTERNAL AUTHENTICATE tested using invalid parameters P1-P2 has been verified.
Status: Test 8.6 Passed."

Case 1.2: If the READ BINARY returns SW1 SW2 == 90 00, then print

"EXTERNAL AUTHENTICATE tested using invalid parameters P1-P2 has not been verified because a subsequent READ BINARY was successful.
Status: Test 8.6 Failed."

Case 2: If the EXTERNAL AUTHENTICATE APDU does not return SW1 SW2 == 6A 86 or 6A 88, then print

"EXTERNAL AUTHENTICATE tested using invalid parameters P1-P2 did not return the correct status code.
Status: Test 8.6 Failed."

9. GET CHALLENGE

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The master file contains an elementary file whose file ID is _goodEfileID4. All services of this file have access control rule ACR_XAUTH.

3. Select the master file:

    Issue the following SELECT MASTER FILE APDU:

        00 A4 03 0C|02|3F00| |

    Case 1: If the SELECT MASTER FILE APDU returns SW1 SW2 == 90 00 or 61 XX, then continue with Step 4.

    Case 2: If the SELECT MASTER FILE APDU does not return SW1 SW2 == 90 00 or 61 XX, then print
    "The master file cannot be selected. GET CHALLENGE cannot be tested."
    End current test.

4. Select the elementary file whose file ID is _goodEfileID4:

    Issue the following SELECT EF UNDER SELECTED DF APDU:

        00 A4 02 0C|02|_goodEFileID4| |

    Case 1: If the SELECT EF UNDER SELECTED DF APDU returns SW1 SW2 == 90 00 or 61 XX, then continue with Step 2 of the Instantiation Scenario.

    Case 2: If the SELECT EF UNDER SELECTED DF APDU does not return SW1 SW2 == 90 00 or 61 XX, then print
    "The elementary file cannot be selected. GET CHALLENGE cannot be tested."
    End current test.

Test for Assertion 9.1

The APDU is tested using valid parameters.

Instantiation Scenario

1. (Pre) Construct the starting state for GET CHALLENGE.

2. Issue the following GET CHALLENGE APDU:

        00 84 00 00| | |08|

Verification Goal:
To verify the expected results:

1. The APDU returns
    - SW1 SW2 == 90 00 (successful execution)

- the 8 byte data field in the response message == a cryptographic challenge.

Perform this verification by inspection.

Verification and Reporting Scenario:

1. **Case 1**: If the GET CHALLENGE APDU returns SW1 SW2 == 90 00, then manually inspect the 8 byte data field in the response message.

    Case 1.1: If the data field contains a valid 8 byte cryptographic challenge, then print
    "GET CHALLENGE tested using valid parameters has been verified.
    Status: Test 9.1 Passed."

    Case 1.2: If the data field does not contain a valid 8 byte cryptographic challenge, then print
    "GET CHALLENGE tested using valid parameters has not been verified by inspection.
    Status: Test 9.1 Failed."

    Case 2: If the GET CHALLENGE APDU does not return SW1 SW2 == 90 00, then print
    "GET CHALLENGE tested using valid parameters did not return the correct status code.
    Status: Test 9.1 Failed."

Test for Assertion 9.2

The APDU is tested in the case where the function is not supported.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 9.3

The APDU is tested using invalid parameters P1-P2.

Instantiation Scenario

1. (Pre) Construct the starting state for GET CHALLENGE.

2. Issue the following GET CHALLENGE APDU:

    00 84 00 FE| | |08|

Verification Goal:

To verify the expected results:

1. The APDU returns
    - SW1 SW2 == 6A 86 (Invalid parameters P1-P2).

Verification and Reporting Scenario:

Case 1: If the GET CHALLENGE APDU returns SW1 SW2 == 6A 86, then print

"GET CHALLENGE tested using invalid parameters P1-P2 has been verified.
Status: Test 9.3 Passed."

Case 2: If the GET CHALLENGE APDU does not return SW1 SW2 == 6A 86, then print

"GET CHALLENGE tested using invalid parameters P1-P2 did not return the correct status code.
Status: Test 9.3 Failed."

10. INTERNAL AUTHENTICATE

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader. The card is subject to internal authentication.

2. The master file contains an elementary file whose file ID is _goodEfileID2. This file is subject to PIN authentication for all services; the PIN is _goodPIN.

3. Select the master file:

   Issue the following SELECT MASTER FILE APDU:

   00 A4 03 0C|02|3F00| |

   Case 1: If the SELECT MASTER FILE APDU returns SW1 SW2 == 90 00 or 61 XX, then continue with Step 2 of the Instantiation Scenario.

   Case 2: If the SELECT MASTER FILE APDU does not return SW1 SW2 == 90 00 or 61 XX, then print
   "The master file cannot be selected. INTERNAL AUTHENTICATE cannot be tested."
   End current test.

Test for Assertion 10.1

The APDU is tested using valid parameters.

Instantiation Scenario

1. (Pre) Construct the starting state for INTERNAL AUTHENTICATE.

2. (Pre) Issue the following VERIFY APDU:

   00 20 00 00|08|_goodPIN| |

   Case 1: If the VERIFY APDU returns SW1 SW2 == 90 00, continue with step 2 of the Instantiation Scenario.

   Case 2: If the VERIFY APDU does not return SW1 SW2 == 90 00, then print
   "VERIFY was not successful. INTERNAL AUTHENTICATE cannot be tested."

   End current test.

3. Issue the following INTERNAL AUTHENTICATE APDU:

   00 88 00 00 08 _goodChallenge 08

Verification Goal:
To verify the expected results:

1. The APDU returns either
   - SW1 SW2 == 90 00 (Successful execution)
   - the 8 byte data field in the response message == the correctly encrypted challenge

   or

- SW1 SW2 == 61 08 (normal processing, data still available)
- the correctly encrypted challenge is available to be retrieved via GET RESPONSE.

Note: For systems using the T=0 communications protocol, the 61 XX response code is required by ISO 7816-4.

Perform this verification by inspection.

Verification and Reporting Scenario:

1. **Case 1**: If the INTERNAL AUTHENTICATE APDU returns SW1 SW2 == 90 00, then manually inspect the 8 byte data field in the response message.

   Case 1.1: If the data field contains the correct 8 byte cryptogram, then print
   "INTERNAL AUTHENTICATE tested using valid parameters has been verified.
   Status: Test 10.1 Passed."

   Case 1.2: If the data field does not contain the correct 8 byte cryptogram, then print
   "INTERNAL AUTHENTICATE tested using valid parameters has not been verified by inspection.
   Status: Test 10.1 Failed."

   Case 2: If the INTERNAL AUTHENTICATE APDU returns the response code 61 08, then issue the following GET RESPONSE APDU
   00 C0 00 00| | |08|

   Case 2.1: If the GET RESPONSE APDU returns
   - SW1 SW2 == 90 00
   - the 8 byte field in the response message contains the correct 8 byte cryptogram

   then print
   "INTERNAL AUTHENTICATE tested using valid parameters has been verified.
   Status: Test 10.1 Passed."

   Case 2.2: If the GET RESPONSE APDU returns
   - SW1 SW2 /= 90 00

   or
   - the 8 byte field in the response message does not contain the correct 8 byte cryptogram

   then print
   "INTERNAL AUTHENTICATE tested using valid parameters has not been verified because a subsequent GET RESPONSE did not return the correct data value.
   Status: Test 10.1 Failed."

   Case 3: If the INTERNAL AUTHENTICATE APDU does not return SW1 SW2 == 90 00 or SW1 SW2 == 61 08, then print
   "INTERNAL AUTHENTICATE tested using valid parameters did not return the correct status code.
   Status: Test 10.1 Failed."

Test for Assertion 10.2

The APDU is tested in the case where the referenced data is deactivated.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 10.3

The APDU is tested in a context where the command is not allowed.

Instantiation Scenario

1. (Pre) Construct the starting state for EXTERNAL AUTHENTICATE.

Note: VERIFY is not issued.

2. Issue the following INTERNAL AUTHENTICATE APDU:
   00 88 00 00 08 _goodChallenge 08

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 69 85 (Conditions of use not satisfied)

Verification and Reporting Scenario:

1. **Case 1**: If the INTERNAL AUTHENTICATE APDU returns SW1 SW2 == 69 85, then print
   "INTERNAL AUTHENTICATE, tested in a context where the command is not allowed, has been verified.
   Status: Test 10.3 Passed."

   Case 2: If the INTERNAL AUTHENTICATE APDU does not return SW1 SW2 == 69 85, then print
   "INTERNAL AUTHENTICATE, tested in a context where the command is not allowed, did not return the correct status code.
   Status: Test 10.3 Failed."

Test for Assertion 10.4

The APDU is tested using invalid parameters P1-P2.

Instantiation Scenario

1. (Pre) Construct the starting state for INTERNAL AUTHENTICATE.

2. (Pre) Issue the following VERIFY APDU:
   00 20 00|08|_goodPIN| |

   Case 1: If the VERIFY APDU returns SW1 SW2 == 90 00, continue with step 2 of the Instantiation Scenario.

   Case 2: If the VERIFY APDU does not return SW1 SW2 == 90 00, then print

"VERIFY was not successful. INTERNAL AUTHENTICATE cannot be tested."

End current test.

3. Issue the following INTERNAL AUTHENTICATE APDU:

   00 88 00 FE|08|_goodChallenge|08|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (Incorrect parameters P1-P2) or 6A 88 (Referenced data not found).

Verification and Reporting Scenario:

1. **Case 1**: If the INTERNAL AUTHENTICATE APDU returns SW1 SW2 == 6A 86 or 6A 88, then print

   "INTERNAL AUTHENTICATE, tested using invalid parameters P1-P2, has been verified.
   Status: Test 10.4 Passed."

   Case 2: If the INTERNAL AUTHENTICATE APDU does not return SW1 SW2 == 69 85, then print

   "INTERNAL AUTHENTICATE, tested using invalid parameters P1-P2, did not return the correct status code.
   Status: Test 10.4 Failed."

11. VERIFY

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The master file contains an elementary file whose file ID is _goodEfileID2. This file is subject to PIN authentication for all services. The PIN is _goodPIN.

3. Select the master file:

   Issue the following SELECT MASTER FILE APDU:

   ```
   00 A4 03 0C|02|3F00| |
   ```

   Case 1: If the SELECT MASTER FILE APDU returns SW1 SW2 == 90 00 or 61 XX, then continue with Step 4.

   Case 2: If the SELECT MASTER FILE APDU does not return SW1 SW2 == 90 00 or 61 XX, then print
   "The master file cannot be selected. VERIFY cannot be tested."
   End current test.

4. Select the elementary file whose file ID is _goodEFileID2:

   Issue the following SELECT EF UNDER SELECTED DF APDU:

   ```
   00 A4 02 0C|02|_goodEFileID2| |
   ```

   Case 1: If the SELECT EF UNDER SELECTED DF APDU returns SW1 SW2 == 90 00 or 61 XX, then continue with Step 2 of the Instantiation Scenario.

   Case 2: If the SELECT EF UNDER SELECTED DF APDU does not return SW1 SW2 == 90 00 or 61 XX, then print
   "The elementary file cannot be selected. VERIFY cannot be tested."
   End current test.

5. Verify that the selected file does not allow read access:

   Issue the following READ BINARY APDU:
   READ BINARY APDU:

   ```
   00 B0 00 02| | |04|
   ```

   Case 1: If the READ BINARY returns SW1 SW2 == 69 82, then continue with Step 6.

   Case 2: If the READ BINARY does not return SW1 SW2 == 69 82, then print
   "It cannot be verified that the currently selected file does not allow read access. VERIFY cannot be tested."
   End current test.

Test for Assertion 11.1

The APDU is tested using valid parameters.

Instantiation Scenario

1. (Pre) Construct the starting state for VERIFY.

2. Issue the following VERIFY APDU:

   00 20 00 00|08|_goodPIN| |

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution).

2. The currently selected file is now open to read access.

Verification and Reporting Scenario:

1. **Case 1**: If the VERIFY APDU returns SW1 SW2 == 90 00, then issue the following READ BINARY APDU:

   00 B0 00 02| | |04|

   Case 1.1: If the READ BINARY returns SW1 SW2 == 90 00, then print

   "VERIFY tested using valid parameters has been verified.
   Status: Test 11.1 Passed."

   Case 1.2: If the READ BINARY does not return SW1 SW2 == 90 00, then print

   "VERIFY tested using valid parameters has not been verified because a subsequent READ BINARY was rejected.
   Status: Test 11.1 Failed."

   Case 2: If the VERIFY APDU does not return SW1 SW2 == 90 00, then print

   "VERIFY tested using valid parameters did not return the correct status code.
   Status: Test 11.1 Failed."

Test for Assertion 11.2

The APDU is tested using invalid authentication data.

Instantiation Scenario

1. (Pre) Construct the starting state for VERIFY.

2. Issue the following VERIFY APDU:

   00 20 00 00|08|_badPIN| |

Verification Goal:

To verify the expected results:

1. The APDU returns

- SW1 SW2 == 63 00 (Verification failed), 63 CX (Verification failed), or 69 83 (Authentication method blocked).

2. The currently selected file is still not open to read access.

<u>Verification and Reporting Scenario</u>:

1. **Case 1**: If the VERIFY APDU returns SW1 SW2 == 63 00, 63 CX, or 69 83, then issue the following READ BINARY APDU:

   ```
   00 B0 00 02|  |  |04|
   ```

   Case 1.1: If the READ BINARY does not return SW1 SW2 == 6982 (security status not satisfied), then print
   "VERIFY tested using invalid authentication data has been verified.
   Status: Test 11.2 Passed."

   Case 1.2: If the READ BINARY returns SW1 SW2 == 90 00, then print
   "VERIFY tested using invalid authentication data has not been verified because a subsequent READ BINARY was successful.
   Status: Test 11.2 Failed."

Case 2: If the VERIFY APDU does not return SW1 SW2 == 90 00, then print
"VERIFY tested using invalid authentication data did not return the correct status code.
Status: Test 11.2 Failed."

Test for Assertion 11.3

The APDU is tested in the case where the referenced data is deactivated.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 11.4

The APDU is tested using invalid parameters P1-P2.

<u>Instantiation Scenario</u>

1. (Pre) Construct the starting state for VERIFY.

2. Issue the following VERIFY APDU:

   ```
   00 20 00 FE|08|_goodPIN|  |
   ```

<u>Verification Goal</u>:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (Incorrect parameters P1-P2).

2. The currently selected file is still not open to read access.

<u>Verification and Reporting Scenario</u>:

1. **Case 1**: If the VERIFY APDU returns SW1 SW2 == 6A 86 , then issue the following READ BINARY APDU:

```
00 B0 00 02|  |  |04|
```

Case 1.1: If the READ BINARY does not return SW1 SW2 == 90 00, then print

"VERIFY tested using invalid parameters P1-P2 has been verified.
Status: Test 11.4 Passed."

Case 1.2: If the READ BINARY returns SW1 SW2 == 6982 (security status not satisfied), then print

"VERIFY tested using invalid parameters P1-P2 has not been verified because a subsequent READ BINARY was successful.
Status: Test 11.4 Failed."

Case 2: If the VERIFY APDU does not return SW1 SW2 == 6A 86, then print

"VERIFY tested using invalid parameters P1-P2 did not return the correct status code.
Status: Test 11.4 Failed."

12. MANAGE SECURITY ENVIRONMENT

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

Test for Assertion 12.1

The APDU is tested using valid parameters.

Instantiation Scenario

1. (Pre) Construct the starting state for MANAGE SECURITY ENVIRONMENT.

2. Issue the following MANAGE SECURITY ENVIRONMENT APDU:

```
00 22 41 B6|_goodKeyRefInfoLen|_goodKeyRefInfo|  |
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution).

Verification and Reporting Scenario:

1. **Case 1**: If the MANAGE SECURITY ENVIRONMENT APDU returns SW1 SW2 == 90 00, then print
   "MANAGE SECURITY ENVIRONMENT tested using valid parameters has been verified.
   Status: Test 12.1 Passed."

   Case 2: If the MANAGE SECURITY ENVIRONMENT APDU does not return SW1 SW2 == 90 00, then print
   "MANAGE SECURITY ENVIRONMENT tested using valid parameters did not return the correct status code.
   Status: Test 12.1 Failed."

Test for Assertion 12.2

The APDU is tested in the case where the Security Environment cannot be set.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 12.3

The APDU is tested using a bad data field length.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 12.4

The APDU is tested using an invalid or missing tag, length or value in a Control Reference Data Object.

Instantiation Scenario

1. (Pre) Construct the starting state for MANAGE SECURITY ENVIRONMENT.

2. Issue the following MANAGE SECURITY ENVIRONMENT APDU:

   00 22 41 B6|05|_badKeyRefInfo| |

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 80 (Invalid or missing tag, length or value in a Control Reference Data Object).

Verification and Reporting Scenario:

1. **Case 1**: If the MANAGE SECURITY ENVIRONMENT APDU returns SW1 SW2 == 6A 80, then print

   "MANAGE SECURITY ENVIRONMENT tested using an invalid or missing tag, length or value in a Control Reference Data Object has been verified.
   Status: Test 12.4 Passed."

   Case 2: If the MANAGE SECURITY ENVIRONMENT APDU does not return SW1 SW2 == 6A 80, then print

   "MANAGE SECURITY ENVIRONMENT tested using an invalid or missing tag, length or value in a Control Reference Data Object did not return the correct status code.
   Status: Test 12.4 Failed."

Test for Assertion 12.5

The APDU is tested using incorrect parameters P1-P2.

Instantiation Scenario

1. (Pre) Construct the starting state for MANAGE SECURITY ENVIRONMENT.

2. Issue the following MANAGE SECURITY ENVIRONMENT APDU:

   00 22 00 FE|_goodKeyRefInfoLen|_goodKeyRefInfo| |

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (Incorrect parameters P1-P2).

Verification and Reporting Scenario:

1. **Case 1**: If the MANAGE SECURITY ENVIRONMENT APDU returns SW1 SW2 == 6A 86, then print

   "MANAGE SECURITY ENVIRONMENT tested using incorrect parameters P1-P2 has been verified.
   Status: Test 12.5 Passed."

Case 2: If the MANAGE SECURITY ENVIRONMENT APDU does not return SW1 SW2 == 6A 86, then print

"MANAGE SECURITY ENVIRONMENT tested using incorrect parameters P1-P2 did not return the correct status code.
Status: Test 12.5 Failed."

13. PERFORM SECURITY OPERATION

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

Test for Assertion 13.1

The APDU is tested using valid parameters.

Instantiation Scenario

1. (Pre) Construct the starting state for PERFORM SECURITY OPERATION.

2. (Pre) Issue the following MANAGE SECURITY ENVIRONMENT APDU:

   00 22 41 B6|_goodKeyRefInfoLen|_goodKeyRefInfo| |

   Case 1: If the MANAGE SECURITY ENVIRONMENT APDU returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the MANAGE SECURITY ENVIRONMENT APDU does not return SW1 SW2 == 90 00, then print
   "MANAGE SECURITY ENVIRONMENT cannot be performed. PERFORM SECURITY OPERATION cannot be tested."
   End current test.

3. Issue the following PERFORM SECURITY OPERATION APDU:

   00 2A 9E 9A|08|_goodMessageDigest|08|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution)
   - the 8 byte data field in the response message == a signed message digest

   or
   - SW1 SW2 == 61 08 (normal processing, data still available)
   - the 8 bytes specified to be read are available to be retrieved via GET RESPONSE.

Note: For systems using the T=0 communications protocol, the 61 XX response code is required by ISO 7816-4.

Perform this verification by inspection.

Verification and Reporting Scenario:

1. **Case 1**: If the PERFORM SECURITY OPERATION APDU returns SW1 SW2 == 90 00, then manually inspect the 8 byte data field in the response message.

   Case 1.1: If the data field contains a valid signed message digest, then print
   "PERFORM SECURITY OPERATION tested using valid parameters has been verified.
   Status: Test 13.1 Passed."

Case 1.2: If the data field does not contain a valid signed message digest, then print

"PERFORM SECURITY OPERATION tested using valid parameters has not been verified by inspection.
Status: Test 13.1 Failed."

Case 2: If the PERFORM SECURITY OPERATION APDU returns the response code 61 08, then issue the following GET RESPONSE APDU

00 C0 00 00| | |08|

Case 2.1: If the GET RESPONSE APDU returns

- SW1 SW2 == 90 00
- the 8 byte field in the response message contains a valid signed message digest

then print

"PERFORM SECURITY OPERATION tested using valid parameters has been verified.
Status: Test 13.1 Passed."

Case 2.2: If the GET RESPONSE APDU returns

- SW1 SW2 /= 90 00

or

- the 8 byte field in the response message does not contain a valid signed message digest

then print

"PERFORM SECURITY OPERATION tested using valid parameters has not been verified because a subsequent GET RESPONSE did not return the correct data value.
Status: Test 13.1 Failed."

Case 3: If the PERFORM SECURITY OPERATION APDU does not return SW1 SW2 == 90 00 or SW1 SW2 == 61 08, then print

"PERFORM SECURITY OPERATION tested using valid parameters did not return the correct status code.
Status: Test 13.1 Failed."

Test for Assertion 13.2

The APDU is tested using an invalid L_C.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 13.3

The APDU is tested using an invalid file type.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 13.4

The APDU is tested where the previous operation was not MANAGE SECURITY ENVIRONMENT.

Instantiation Scenario

1. (Pre) Construct the starting state for PERFORM SECURITY OPERATION.

2. Issue the following PERFORM SECURITY OPERATION APDU:

   00 2A 9E 9A|08|_goodMessageDigest|08|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 69 85 (No preceding MSE-Set).

Verification and Reporting Scenario:

1. **Case 1**: If the PERFORM SECURITY OPERATION APDU returns SW1 SW2 == 69 85, then print
   "PERFORM SECURITY OPERATION tested where the previous operation was not MANAGE SECURITY ENVIRONMENT has been verified.
   Status: Test 13.4 Passed."

   Case 2: If the PERFORM SECURITY OPERATION APDU does not return SW1 SW2 == 69 85, then print
   "PERFORM SECURITY OPERATION tested where the previous operation was not MANAGE SECURITY ENVIRONMENT did not return the correct status code.
   Status: Test 13.4 Failed."

Test for Assertion 13.5

The APDU is tested with a missing message digest.

Instantiation Scenario

1. (Pre) Construct the starting state for PERFORM SECURITY OPERATION.

2. (Pre) Issue the following MANAGE SECURITY ENVIRONMENT APDU:

   00 22 41 B6|_goodKeyRefInfoLen|_goodKeyRefInfo| |

   Case 1: If the MANAGE SECURITY ENVIRONMENT APDU returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the MANAGE SECURITY ENVIRONMENT APDU does not return SW1 SW2 == 90 00, then print
   "MANAGE SECURITY ENVIRONMENT cannot be performed. PERFORM SECURITY OPERATION cannot be tested."
   End current test.

3. Issue the following PERFORM SECURITY OPERATION APDU:

   00 2A 9E 9A|00| | |

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 69 87 (Missing Secure Data Messaging Object).

Verification and Reporting Scenario:

1. **Case 1**: If the PERFORM SECURITY OPERATION APDU returns SW1 SW2 == 69 87, then print
   "PERFORM SECURITY OPERATION tested with a missing message digest has been verified.
   Status: Test 13.5 Passed."

   Case 2: If the PERFORM SECURITY OPERATION APDU does not return SW1 SW2 == 69 87, then print
   "PERFORM SECURITY OPERATION tested with a missing message digest did not return the correct status code.
   Status: Test 13.5 Failed."

Test for Assertion 13.6

The APDU is tested with an invalid message digest.

Instantiation Scenario

1. (Pre) Construct the starting state for PERFORM SECURITY OPERATION.

2. (Pre) Issue the following MANAGE SECURITY ENVIRONMENT APDU:

   ```
   00 22 41 B6|_goodKeyRefInfoLen|_goodKeyRefInfo|  |
   ```

   Case 1: If the MANAGE SECURITY ENVIRONMENT APDU returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the MANAGE SECURITY ENVIRONMENT APDU does not return SW1 SW2 == 90 00, then print
   "MANAGE SECURITY ENVIRONMENT cannot be performed. PERFORM SECURITY OPERATION cannot be tested."
   End current test.

3. Issue the following PERFORM SECURITY OPERATION APDU:

   ```
   00 2A 9E 9A|01|FF|00
   ```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 69 88 (Incorrect Secure Data Messaging Object).

Verification and Reporting Scenario:

1. **Case 1**: If the PERFORM SECURITY OPERATION APDU returns SW1 SW2 == 69 88, then print
   "PERFORM SECURITY OPERATION tested with an invalid message digest has been verified.
   Status: Test 13.6 Passed."

   Case 2: If the PERFORM SECURITY OPERATION APDU does not return SW1 SW2 == 69 88, then print
   "PERFORM SECURITY OPERATION tested with an invalid message digest did not return the correct status code.
   Status: Test 13.6 Failed."

Test for Assertion 13.7

The APDU is tested using invalid parameters P1-P2.

Instantiation Scenario

1. (Pre) Construct the starting state for PERFORM SECURITY OPERATION.

2. (Pre) Issue the following MANAGE SECURITY ENVIRONMENT APDU:

   00 22 41 B6|05|_goodKeyRefInfo| |

   Case 1: If the MANAGE SECURITY ENVIRONMENT APDU returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the MANAGE SECURITY ENVIRONMENT APDU does not return SW1 SW2 == 90 00, then print
   "MANAGE SECURITY ENVIRONMENT cannot be performed. PERFORM SECURITY OPERATION cannot be tested."
   End current test.

3. Issue the following PERFORM SECURITY OPERATION APDU:

   00 2A 00 FE|08|_goodMessageDigest|08|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (Incorrect parameters P1-P2).

Verification and Reporting Scenario:

1. **Case 1**: If the PERFORM SECURITY OPERATION APDU returns SW1 SW2 == 6A 86, then print
   "PERFORM SECURITY OPERATION tested using invalid parameters P1-P2 has been verified.
   Status: Test 13.7 Passed."

   Case 2: If the PERFORM SECURITY OPERATION APDU does not return SW1 SW2 == 6A 86, then print
   "PERFORM SECURITY OPERATION tested using invalid parameters P1-P2 did not return the correct status code.
   Status: Test 13.7 Failed."

Test for Assertion 13.8

The APDU is tested using an invalid length for the response string.

Instantiation Scenario

1. (Pre) Construct the starting state for PERFORM SECURITY OPERATION.

2. (Pre) Issue the following MANAGE SECURITY ENVIRONMENT APDU:

   00 22 41 B6|05|_goodKeyRefInfo| |

   Case 1: If the MANAGE SECURITY ENVIRONMENT APDU returns SW1 SW2 == 90 00, then continue with Step 3.

Case 2: If the MANAGE SECURITY ENVIRONMENT APDU does not return SW1 SW2 == 90 00, then print

"MANAGE SECURITY ENVIRONMENT cannot be performed. PERFORM SECURITY OPERATION cannot be tested."

End current test.

3. Issue the following PERFORM SECURITY OPERATION APDU:

   00 2A 9E 9A|08|_goodMessageDigest|01|

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6C 08 (Wrong length).

Verification and Reporting Scenario:

1. **Case 1**: If the PERFORM SECURITY OPERATION APDU returns SW1 SW2 == 6C 08, then print

   "PERFORM SECURITY OPERATION tested using an invalid length for the response string has been verified.
   Status: Test 13.8 Passed."

   Case 2: If the PERFORM SECURITY OPERATION APDU does not return SW1 SW2 == 6C 08, then print

   "PERFORM SECURITY OPERATION tested using an invalid length for the response string did not return the correct status code.
   Status: Test 13.8 Failed."

Appendix A

List of Symbolic Constants

_badCryptogram == 00 00 00 00 00 00 00 00

_badDFileID == FE 01

_badEfileID == FE 02

_badKeyRefInfo == 00 00 00 00 00

_badKeyRefInfoLen == 05

_badPIN == 00 00 00 00 00 00 00 00

_badPINLen = 08

_existingValue1 == 01 01 01 01

_existingValue2 == 02 02 02 02

_existingValue3 == 03 03 03 03

_existingValue4 == 04 04 04 04

_existingValue5 == 05 05 05 05

_existingValue6 == 06 06 06 06

_existingValue7 == 07 07 07 07

_goodChallenge == 01 02 03 04 05 06 07 08

_ goodChallengeLen = 08

_goodDFileID == 01 01

_goodEFileID1 == 00 01

_goodEFileID2 == 00 02

_goodEFileID3 == 00 03

_goodEFileID4 == 00 04

_goodKeyRefInfo == 84 01 01 91 00

_goodKeyRefInfoLen == 05

_goodMessageDigest == 08 07 06 05 04 03 02 01

_ goodMessageDigestLen = 08

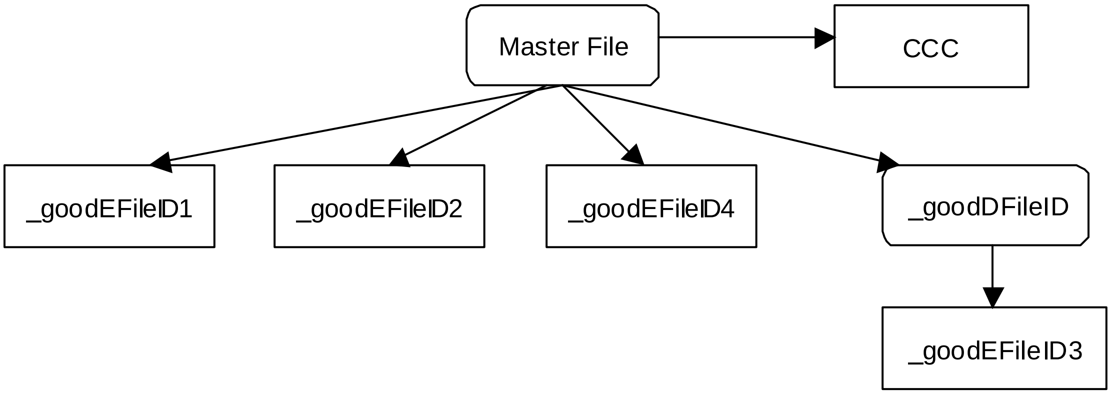
_goodPIN == 00 01 00 02 00 03 00 04

```
_goodPINLen = 08

_newValue1 == 11 11 11 11

_newValue2 == 22 22 22 22

_newValue3 == 33 33 33 33

_newValue4 == 44 44 44 44

_newValue5 == 55 55 55 55
```

Appendix B

Description of the Card Required for Testing VCEI (File System) Implementations

The required card has the following container structure:



1. Description of the elementary file with ID _goodEFileID1:

   - All services of this file have access control rule ACR_ALWAYS.

   - The file contains only the following four TLV records:
     - byte 00: 01
       byte 01: 04
       bytes 02-05: _existingValue1
     - byte 06: 02
       byte 07: 04
       bytes 08-0B: _existingValue2
     - byte 0C: 03
       byte 0D: 04
       bytes 0E-11: _existingValue3
     - byte 12: 04
       byte 13: 04
       bytes 14-17: _existingValue4.

   - The file has a maximum size of 24 bytes.

   - The file does not have a special length TLV (section 8.3 of the Spec.)

2. Description of the elementary file with ID _goodEFileID2:

   - All services of this file have access control rule ACR_PIN, with the PIN == _goodPIN.

- The file contains only the following TLV record:
  - byte 00: 11
    byte 01: 04
    bytes 02-05: _existingValue5.

- The file does not have a special length TLV.

3. Description of the elementary file with ID _goodEFileID3:

   - All services of this file have access control rule ACR_XAUTH.

   - The file contains the following two TLV records:
     - byte 00: 21
       byte 01: 04
       bytes 02-05 == _existingValue6
     - byte 06: 22
       byte 07: 04
       bytes 08-0B == _existingValue7.

   - The file does not have a special length TLV.

4. Description of the elementary file with ID _goodEFileID4:

   - All services of this file have access control rule ACR_XAUTH.

   - The file contains the following TLV record:
     - byte 00: 41
       byte 01: 04
       bytes 02-05: _existingValue8.

   - The file does not have a special length TLV.

5. The card does not contain a dedicated file with file ID _badDFileID directly under the Master File.

6. The card does not contain an elementary file with file ID _badEFileID.

Substantive changes since the March 31 version of this document

1. All SELECT APDUs now allow the response code 61 XX as a correct response wherever they allow 90 00.

2. In Section 1, it is now made explicit that we assume that LL < FF.

3. Section 1.2 now uses L_e == 1 in the GET RESPONSE in the Instantiation Scenario.

4. Sections 1.5, 4.7, 5.9, and 6.6 now use P1 P2 == 00 FE in the SELECT in the Instantiation Scenario.

5. In Sections 4.1 and 4.2, the offsets in the READ BINARYs in the Verification Scenarios have been corrected.

6. In the Starting State for section 7 and in Appendix B, it is now made explicit that the elementary file whose file ID is _goodEfileID2 has a maximum size of 17 bytes.

7. Section 7.2 now uses P2 == 16 in the UPDATE BINARY in the Instantiation Scenario, P2 == 14 in the READ BINARY in the Verification Scenario, and _existingValue4 in the verification test.

8. Section 11.4 now uses P1 P2 == 00 FE in the VERIFY in the Instantiation Scenario.

9. Section 12.5 now uses P1 P2 == 00 FE in the MANAGE SECURITY ENVIRONMENT in the Instantiation Scenario.

10. In Section 13.4, the instruction code for the PERFORM SECURITY OPERATION in the Instantiation Scenario is now == 2A.

11. In Section 13.6, the PERFORM SECURITY OPERATION in the Instantiation Scenario now uses L_c == 01 and Data Field == FF.

12. Section 13.7 now uses P1 P2 == 00 FE in the PERFORM SECURITY OPERATION in the Instantiation Scenario.

13. Section 13.8 now uses Le == 01 in the PERFORM SECURITY OPERATION in the Instantiation Scenario.

14. Appendix A now defines _badCryptogram to be an empty field.